

2. Übung zur Vorlesung

COMPUTERORIENTIERTE MATHEMATIK I

WS 2020/2021

[http://numerik.mi.fu-berlin.de/wiki/WS\\_2020/CoMaI.php](http://numerik.mi.fu-berlin.de/wiki/WS_2020/CoMaI.php)

**Abgabe: Do., 3. Dezember 2020, 12:15 Uhr**

**1. Aufgabe** (4 TP)

Führen Sie die folgenden Rechenaufgaben mit Dualzahlen aus, ohne in das Dezimalsystem umzurechnen:

a)  $0,1100101_2 \cdot 10101,111_2 = ?$

b)  $\frac{10_2}{110_2} + \frac{101_2}{10100_2} = ?$

**2. Aufgabe** (4 TP)

Beweisen oder widerlegen Sie die folgenden Aussagen:

a) Jeder endliche Dualbruch ist auch ein endlicher Dezimalbruch.

b) Jeder endliche Dezimalbruch ist auch ein endlicher Dualbruch.

**3. Aufgabe** (6 PP)

In dieser Aufgabe sollen Sie in PYTHON eine Funktion implementieren, die ganze Zahlen im Dezimalsystem als Zahlen im Dualsystem darstellt. Gehen Sie dabei in mehreren Schritten vor:

a) Implementieren Sie eine Funktion `ntobasetwo(n, c)`, die eine natürliche Zahl  $n \in \mathbb{N}$  in eine Binärzahl der Länge  $c$  umwandelt.

Als Rückgabewert wird eine Liste  $b$  der Länge  $c$  erwartet, sodass

$$n = \sum_{i=0}^{c-1} b_i 2^i$$

sowie  $b_i \in \{0, 1\}$  für alle  $i \in \{0, \dots, c-1\}$  gilt.

Für den Fall, dass  $n$  nicht als Binärzahl der Länge  $c$  dargestellt werden kann, soll Ihr Programm das Ergebnis entsprechend abschneiden. Sprich, falls  $n = \sum_{i=0}^m b_i 2^i$  mit  $m > c$  gilt, sollen nur die  $b_i$  mit  $i \in \{0, \dots, c-1\}$  zurückgegeben werden.

- b) Implementieren Sie eine Funktion `complement(b)`, die das Zweierkomplement einer Binärzahl entsprechend der Vorlesung berechnet. Dabei wird als Eingabe eine Liste  $b$  erwartet mit  $b_i \in \{0, 1\}$ . Der Rückgabewert soll auch eine Liste  $\hat{b}$  mit  $\hat{b}_i \in \{0, 1\}$  sein, sodass  $b$  und  $\hat{b}$  dieselbe Länge haben.
- c) Implementieren Sie eine Funktion `ztobasetwo(z, c)`, die eine ganze Zahl  $z \in \mathbb{Z}$  in eine Binärzahl in Form einer Liste der Länge  $c$  umwandelt, wobei negative Zahlen mit Hilfe des Zweierkomplements realisiert werden. Verwenden Sie dabei die Funktionen aus den vorangegangenen Teilaufgaben.

Dokumentieren und testen Sie Ihren Code!

**Wichtig:** Verwenden Sie keine der in PYTHON (oder Python-Bibliotheken) vordefinierten Funktionen zur direkten Umwandlung zwischen Zahlensystemen! Sie können diese aber nutzen, um Ihren Code zu testen.

**Hinweis:** Eventuell sind die Funktionen `numpy.mod` (oder der `%` Operator), `numpy.floor` oder `numpy.ceil` aus der NUMPY-Bibliothek für Sie hilfreich.

#### 4. Bonusaufgabe (Quiz) (1 Bonus TP/PP)

Formulieren Sie eine Frage zur Vorlesung. Falls Sie die Antwort wissen, geben Sie die richtige Antwort und 3 falsche Antwortmöglichkeiten an.

#### ALLGEMEINE HINWEISE

Die Punkte unterteilen sich in Theoriepunkte (TP) und Programmierpunkte (PP). Bitte beachten Sie die auf der Vorlesungshomepage angegebenen Hinweise zur Bearbeitung und Abgabe der Übungszettel, insbesondere der Programmieraufgaben.